Mohd Ayub A. F. et al. (Eds.) (2011). Workshop Proceedings of the 19th International Conference on Computers in Education. ChiangMai, Thailand: Asia-Pacific Society for Computers in Education.

Incorporating Framing into SQL-Tutor

Moffat MATHEWS & Antonija MITROVIC
Intelligent Computer Tutoring Group, University of Canterbury, New Zealand
moffat.mathews@canterbury.ac.nz
tanja.mitrovic@canterbury.ac.nz

Abstract: In this paper, we present our work on framing with the view of implementing it in an Intelligent Tutoring System (ITS). The process of framing a learning activity, in our case problem solving, consists of having the activity in between a pre-action (or priming) phase and a post-action (or reflective) phase. In previous work, we found that simulated framing, in which the priming and reflection phases were led by a human teacher while the learning activity itself was performed in an ITS, significantly reduces learning time and requires less effort for similar gains. This paper presents the next stage of the project, in which the priming phase is implemented in the ITS. We performed a pilot study using the extended system, which resulted in the same trends as simulated framing.

Keywords: Intelligent tutoring systems, framing, teaching strategies

Introduction

In previous work [1] we have presented the initial results on the framing teaching strategy. Framing is a pedagogical strategy that we have distilled from educators' practices in tertiary institutions and high schools. The strategy consists of three sequential phases whereby the learning activity (action phase) is preceded by a pre-action (or priming) phase and followed by a post-action (or reflection) phase. All three phases together form a learning session. In the classroom, students generally participate in the pre- and post-action phases as a group, while the action phase is done either individually or as a group.

The purpose of the pre-action phase is to prepare the student for the learning activity by helping them focus on the concepts that will be used in the learning activity. The aim of this phase is not to teach them the declarative knowledge required but to "set the scene" for the learning activity. Teachers could lead the short, interactive session by (re-) introducing the target concepts, linking them to previously learned concepts, working through examples, discussing common misconceptions, and setting the "boundaries" for the session.

The learning activity phase immediately follows the pre-action phase. Here, the student takes part in some activity that helps them interact with material relating to the target concepts. For example, students might solve problems, engage in discussion, conduct exploratory research, or run experiments. This phase is self-directed, enabling the student to put into practice what they have learned, and the teacher might provide feedback.

Once the learning activity is complete, the teacher leads the students in the reflection phase. The purpose of this phase is to encourage students to reflect on what they have learned in the previous two phases. Students are encouraged to analyze their errors (including the source of these errors) thereby uncovering misconceptions.

There are several theories that make framing a plausible teaching strategy. Cognitive Load Theory [3] suggests that problem solving for novices generates heavy working memory loads, which could be detrimental to learning. To balance these loads, teachers should

provide guided instruction, help narrow the problem search space by creating "boundaries" to each session, and alleviate the working memory restriction by making sure that only items relevant to the task are loaded into the working memory. This is exactly what happens during the priming phase.

Meta communication about the presentation of the subject is important for learning [4]. Prior to the learning activity, the student needs to know the boundaries of the lesson segment (exactly what the lesson will contain), which should be well defined by the teacher. The student needs to know the content of the session, differentiating between the old and the new material. They also need to know the links between the new knowledge and previously learned knowledge [4].

Many learning models view learning as a cyclic process, around which knowledge acquisition, knowledge application and reflection occur. Andreasen and Wu [5] discuss a few of the commonly used experiential models. Framing is a simplified (and thus possibly easier-to-implement) version of many of the models.

Reflection promotes deep learning [6, 7, 8]. Critically analysing the learning experience helps challenge the student's underlying perception of the domain, identify and correct misconceptions, and integrate new knowledge with existing knowledge [9]. This also allows the student to transfer the newly acquired knowledge to other types of problems or scenarios. Self-explaining one's actions [10] and monitoring one's progress via open student models [11, 12] have been shown to be useful reflective tools that benefit learning.

Our project consists of three main stages:

1. The learning activity is implemented within SQL-Tutor [2], while the pre- and post-action phases are facilitated by a human tutor. The purpose of this stage was to investigate the potential of framing to improve learning before actually implementing it in the ITS. The results of this stage were presented in [1] are reviewed in Section 1.
2. The pre-action and learning activity phases are implemented in SQL-Tutor. This is the current stage of our project.
3. The reflection phase is also implemented in the ITS.

We chose a suite of metrics early on to validate whether the way in which we implemented Framing helps to achieve the intent. These metrics include Learning Efficiency [13], help-usage metrics e.g. High-Level Help, Requests for Help [14, 15], meta-data about problems solved and problems attempted (including difficulty levels), learning curves, and pre and post-tests. The pre and post-tests were designed by a teacher to measure students' knowledge. The same pre and post-tests are to be used in all stages.

1. Stage 1: Simulating the Framing Strategy

As stated earlier, the purpose of this stage was to simulate the Framing strategy in the manner in which we planned to implement it in the ITS. This helped us gather some information about framing with regards to learning and test out our decisions prior to implementation. We selected a set of target SQL concepts, namely the concepts covered by queries using the *Group By* and *Having* clauses, which students generally find difficult to learn. SQL-Tutor was restricted to only present problems relating to these target concepts. The study was held immediately after the relevant concepts had been covered in lectures. The learning activity was problem solving in SQL-Tutor. The pre- and post-action phase were interactive, whiteboard, group sessions, led by a human teacher. The pre-action and post-action phases were limited to 10 minutes each, while the whole session lasted 100 minutes. In the pre-action phase, the teacher briefly reminded students of the target concepts (taught in lectures previously) and, eliciting student participation, worked through a few

examples of varying difficulty. The teacher also discussed typical misconceptions. After interacting with SQL-Tutor, the students were prompted to reflect on their learning experience by commenting on some of their own mistakes. The teacher also showed them the most common mistakes that are usually made during the problem-solving phase. Students were asked to find the errors (in terms of concepts and methods) in those incorrect solutions before collectively working through to reach a correct solution.

Thirty-eight students from a second year database course participated in the evaluation for no monetary reward. We divided participants randomly into two groups: experimental and control. The idea was to perform the evaluation in a setting that was as close to the normal learning environment faced by students. As such, the experimental and control sessions were held during regular course lab sessions. Students in both groups completed a pre-test and a post-test, which were of comparable difficulty and contained three questions relating to the target concepts with the maximum mark of 12. After the pre-test, the experimental group went through priming, followed by problem-solving and reflection phases, which were run as described. In contrast, the control group entered the problem-solving phase immediately after the pre-test. The pre-test, post-test, and problem-solving phases for both groups were identical.

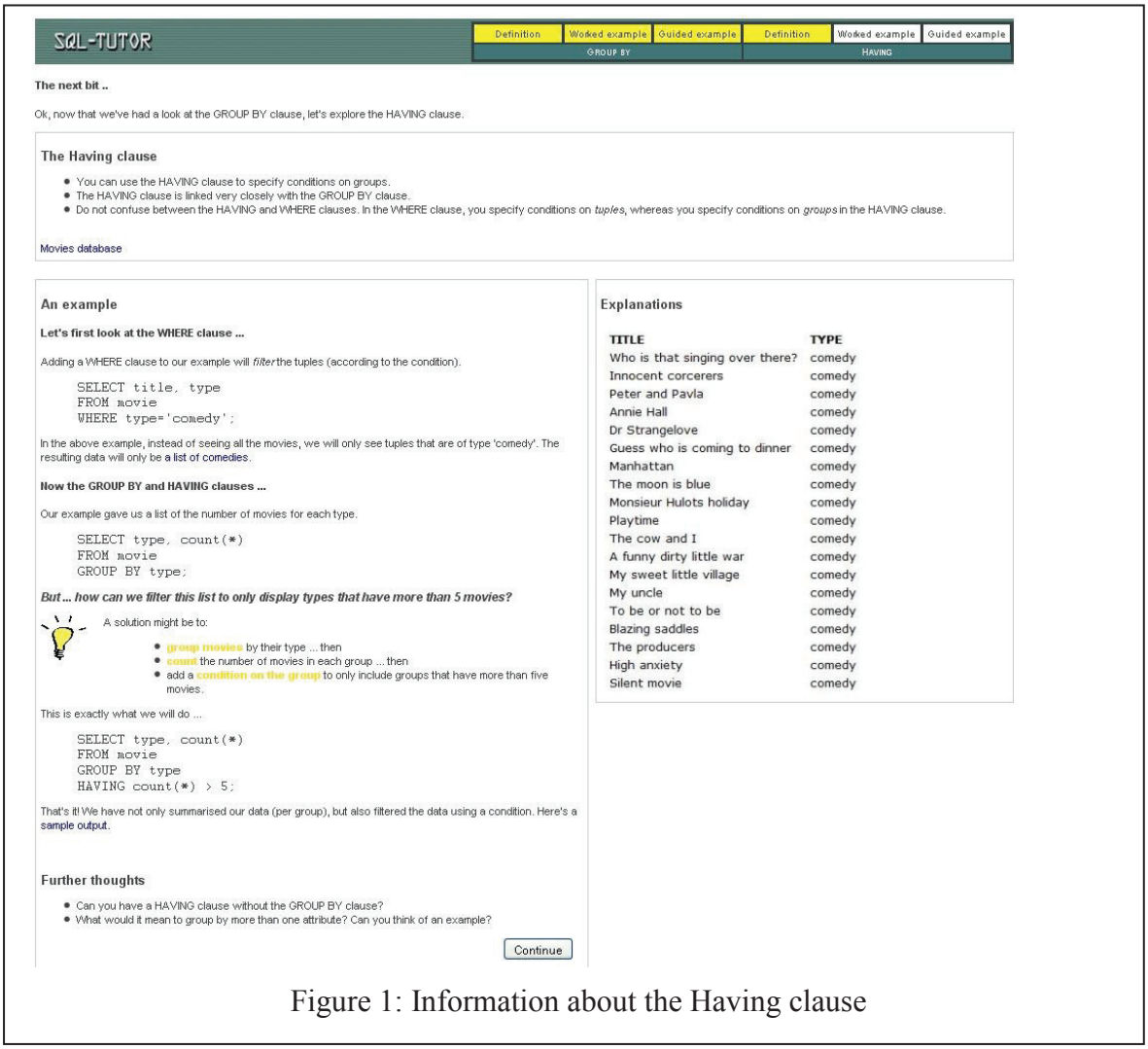

Figure 1: Information about the Having clause

The results of this preliminary study [1] showed that the experimental group had a higher problem-solving speed even though they attempted and solved problems of similar difficulty while using similar levels of help. Furthermore, the experimental group was significantly more efficient in their problem-solving phase than the control group. In other

words, while they did not learn more than the control group, they expended significantly less effort and therefore were more efficient.

2. Implementing Priming

We implemented the priming (pre-action) phase within the ITS using the lessons learned from stage 1. The design of this stage was similar, except that we excluded the post-action phase. The pre-action phase contained three steps for each of the target clauses (i.e. three for the Group By clause followed by three for Having clause). Each of the three steps increased from passive to more active in terms of student interaction. The first step contained the declarative knowledge about the clause followed by an example (see Figure 1). The example provided detailed explanation on how to solve the problem, and a possible solution. Students could also click on a link to display the result of the query. A "Further thoughts" section at the end gave more information to extend their knowledge of the clause. Once the student read the information on this page, they proceeded to the next step.

The second (and fifth) step contained a worked example. Students could hover over parts of the solution to get a detailed explanation for that part of the solution. Figure 2 shows the worked example and the explanation for the condition in the Having clause. Hovering over each part of the solution also highlighted the relevant part of the problem statement, allowing students to link the problem to the solution. Students could also click on a link to view the intended output of the query, or view the database schema.

The third (and sixth) step contained a strictly guided example. Similar to step 2, this step contained a problem statement and an empty solution statement (with blanks that the student had to fill in). When the student clicked on one of the blanks, the explanation for solving that part of the problem was displayed. Figure 3 shows the situation when the student asked for the explanation for the blank in the Having clause. The student could click the "Check" button to check their solution. If the student made an error on one of the parts of the solution, the explanation also contained a bottom out hint telling the student what to enter.

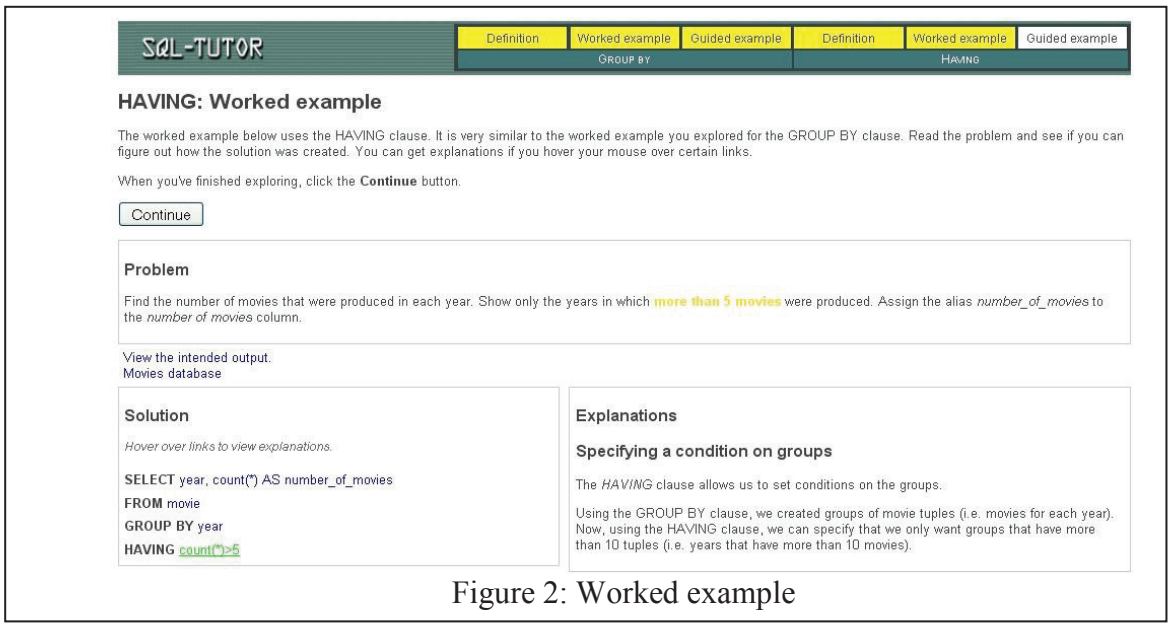

Figure 2: Worked example

The learning activity (problem-solving) immediately followed the six steps of the pre-action phase. This phase was identical to stage 1, where students worked on problems in SQL-Tutor. The problem set was restricted to problems using the target concepts.

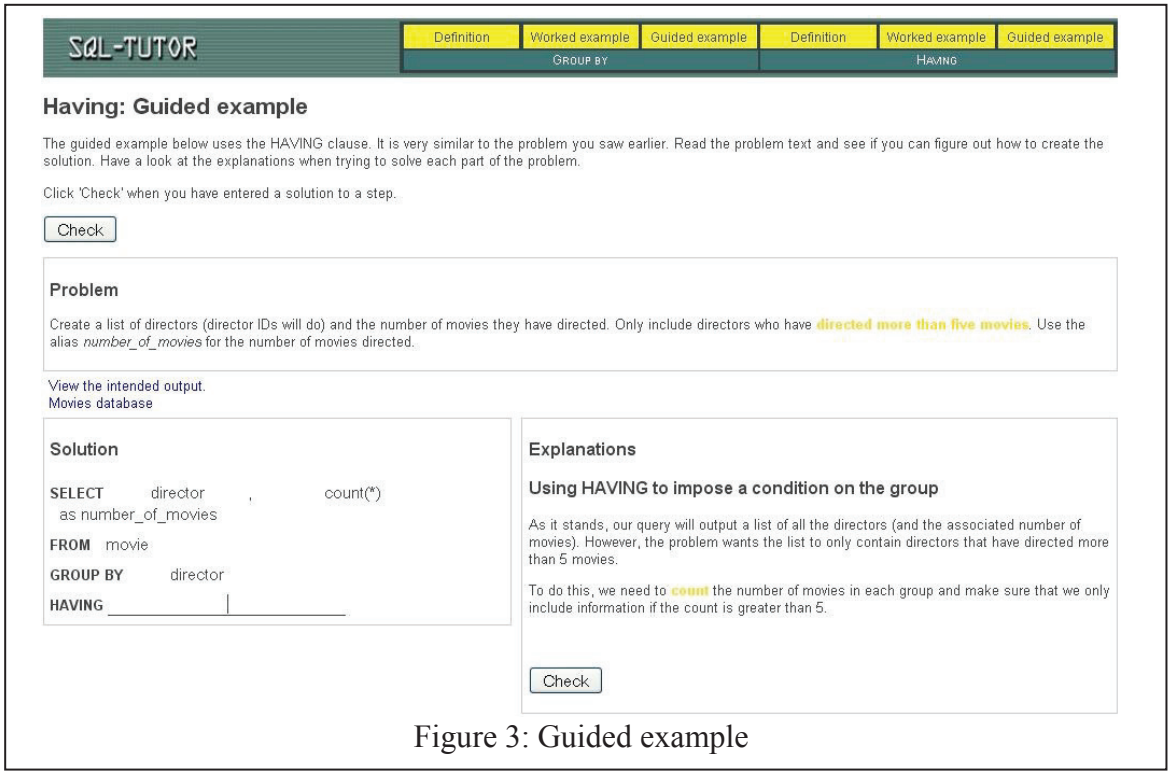

Figure 3: Guided example

3. Results

Thirty students participated in the evaluation for no monetary reward. We divided them randomly into two groups: experimental and control. Two sessions were held during regular course lab sessions (100 minutes long) on 13 and 14 May 2009 respectively. The students participated in the study during the lab session they normally attended throughout the course. Students in both groups completed the pre- and post-test (the same ones used in stage 1 of the project). Following the pre-test, the experimental group went through the pre-action phase in the newly added component, while the control group went directly onto the problem-solving phase.

Table 1: Matched means and standard deviations for test scores (%) and gains

	Pre-test	Post-test	Gain
Experimental group (*n*=5)	57.14% (*s.d* = 39.7)	75% (*s.d* = 16.6)	33.3% (*s.d* = 39.5)
Control group (*n*=12)	52.9% (*s.d* = 26.3)	88.3% (*s.d* = 11.2)	36.6% (*s.d* = 24.7)

The data we collected and analyzed included the pre/post-test results and just over 29 hours (total) of SQL-Tutor student models and logs in which 30 students collectively made 1,769 submissions to the system. There were 17 students in the control group and 13 in the experimental. However, only 17 students sat both tests, and we give the matched results in Table 1. There were no significant differences in the performances of the two groups on the pre-tests, post-tests or between gains (the gain is the difference between post- and pre-test score). There was a significant difference between the pre- and post-test performance of each group, indicating that students improved their domain knowledge during the session.

These results provide us with "trends" even with the low number of students from the experimental group that sat both tests (n=5).

The rest of the analyses were carried out on all the thirty students and the results are reported in Table 2. The trends in this stage were very similar to those found in stage 1. The experimental group spent less time solving problems; this was marginally significant ($t(25)=1.3$, $p=.09$). Students in both groups solved a similar number of problems. This means that the experimental group solved problems at a slightly faster rate, which was also marginally significant ($t(19)=0.46$, $p=.09$).

We analyzed the problem difficulty levels for both groups. Did students in one group attempt or solve problems that were significantly more difficult than the other that might account for the differing speed of problem solving? Each problem in SQL-Tutor is assigned a difficulty level by the SQL expert who authored the problems. Difficulty levels range from 1 (easy) to 9 (difficult) with non-trivial differences in difficulty between levels. SQL experts have checked problem difficulty levels such that problems with the same difficulty level are of similar difficulty. The problems attempted and solved were also of similar difficulty between groups. This was also confirmed for the highest and lowest difficultly level of problems attempted and solved in both groups i.e. students solved similar types of problems.

On average, the experimental group made 49 (26.6) attempts while solving problems, while the control group made 68 (49.3) attempts; the difference was not significant showing that the both groups got similar amounts of feedback from the system. However, to check that students from one group did not receive higher levels of feedback (e.g. they used full solution much more than the other group), we calculated the high-level help used for both groups. *High-level help* (HLH) [14, 15] is defined as the type of help given by a system that provides (part or all of) the correct solution to the student rather than having the student to *solve* the problem; e.g. full solution is a type of HLH. Another important characteristic of HLH in SQL-Tutor is that the HLH levels have to be manually requested by the student whereas the ITS might automatically provide other types of feedback (Low-level help). The *HLH ratio* is the number of HLH attempts divided by the total number of attempts. This shows us the proportion of HLH use, from 0 (no HLH use) to 1 (the student used HLH on every attempt). Students from both groups used similar amounts of high-level help during this phase; 0.46 (0.26) for the experimental group and 0.43 (0.34) for the control group.

Table 2: Results for experimental and control groups

	Experimental	**Control**
Difficulty of problems attempted	5.02 (0.59)	4.95 (0.53)
Difficulty of problems solved	5.01 (0.55)	4.91 (0.56)
Lowest difficulty of problems attempted	3.53 (0.51)	3.64 (0.49)
Highest difficulty of problems attempted	7.0 (1.35)	6.82 (1.7)
Lowest difficulty of problems solved	3.61 (0.50)	3.64 (0.49)
Highest difficulty of problems solved	6.92 (1.32)	6.70 (1.82)
Number of problems solved	10.15 (5.03)	10.5 (6.4)
Time spent on problem solving (min)	52.46 (18.09)	65.17 (33.9)
High-level Help (HLH) ratio	0.46 (0.26)	0.43 (0.34)
Request for Help (RFH) attempts	1.84 (0.68)	1.88 (1.40)
Relative learning efficiency (E)	0.11	-0.12

The relative *learning efficiency* (E) is defined as the performance gained in one condition (the experimental condition) over the effort expended in relation to another condition (the control condition). A condition is more efficient if "1) their performance is higher than expected on the basis of their effort and/or 2) their invested effort is lower than might be expected on the basis of their performance" [13]. To calculate the efficiency of problem-solving for each group, we used "time" as the effort spent and "test gains" as the performance measure. The relative efficiency is found by first converting each of the raw

scores to a *z* score by subtracting the grand mean from the raw score and dividing by the standard deviation. E scores then are found by calculating the perpendicular distance between each *z* score and the E=0 line when plotted on a Cartesian graph. As with stage 1, the efficiency of the experimental group (E = 0.11) was higher than that of the control group (E = -0.12). This was marginally significant ($t(28)=1.11$, $p=0.1$, one-tailed, assuming unequal variances).

We also plotted learning curves for both groups (4). Although the differences were not significant, the trend lines indicate that the experimental group learned at a higher rate than the control group.

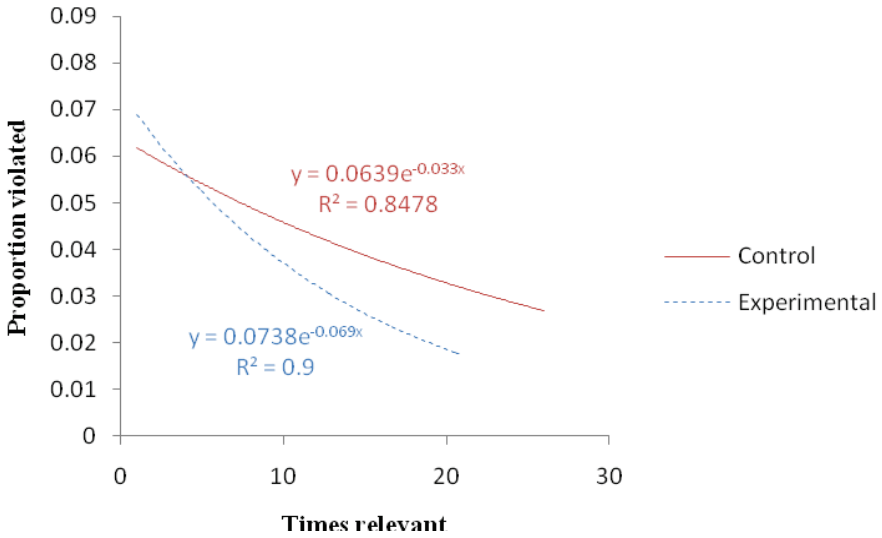

Figure 4: Case study 2: Learning curves for experimental and control groups

4. Discussion and Conclusions

This paper presented the second stage of our project, aiming to implement the framing teaching strategy in an ITS. In previous work, we performed a preliminary study with simulated framing, in which the pre-action and post-action stages were led by a human teachers instead of being implemented in the ITS. The aim of stage 1 was to see whether framing is an effective strategy for an ITS before actually implementing it and therefore committing significant recourses. The results of Stage 1 show that Framing results in significantly faster and more efficient learning.

In this current (second) stage of the project, we implemented the priming phase in SQL-Tutor. This is the first time framing has been implemented in ITSs. The trends gathered from the evaluation of this stage suggest that this implementation worked in a similar manner to that in stage 1. Note that this does not mean that we have achieved an ideal implementation. In fact, although the trends were similar to stage 1, the results gained were not as significant. We have pinpointed at least four possible reasons. First, we had a small number of participants, and therefore cannot make solid conclusions. Secondly, even though the pre-action phase in stage 1 was non-adaptive to the individual, the human teacher adapted to the group as a whole, especially during the worked examples step (when the teacher interacted with the group). This might have increased the effectiveness of the pre-action phase in stage 1. Thirdly, we decided to omit the "common misconceptions" step and only concentrate on correct knowledge. One reason was to keep the pre-action phase reasonably short (to stop it encroaching on the problem-solving). Another reason for the

omission was that presenting correct knowledge followed by incorrect knowledge (common misconceptions) did not seem intuitive using our method of presentation. Finally, the method of presentation differed in both stages. While we had a human teacher (animated, expressive, utilizing the student's visual and auditory senses) presenting in the first stage, we had a series of web pages with limited interaction in the second stage.

The results from this stage, added to that of the previous stage, increase our knowledge and give us a more detailed picture about various decisions we made and aspects of this strategy. Due to the evidence gathered in these stages, it is possible to implement the post-action phase in stage 3 and thus have a system that fully employs the Framing strategy in SQL-Tutor. However, information gathered from this stage suggests that we also could split the development path into a spike that evaluates some of the reasons given in the discussion above and tries to improve on the pre-action phase (say, stage 2B) while continuing development on stage 3. As we have gathered baseline information in stage 2 regarding the pre-action phase, the two stages (stage 2B and stage 3) can be undertaken concurrently. If the spike in stage 2B is successful, the improved pre-action phase can be added with confidence to the system at a later date.

References

[1] Mathews, M., Mitrovic, A. Does framing a problem-solving scenario influence learning? In: Kong, S.C., et al. (Eds.) Proc. 17th Int. Conf. Computers in Education ICCE 2009, Hong Kong: Asia-Pacific Society for Computers in Education, pp. 27-34, 2009.

[2] Mitrovic, A. (2003). An Intelligent SQL Tutor on the Web. *Artificial Intelligence in Education, 13*(2-4), 173-197.

[3] Kirschner, P. A., Clark, R. E., & Sweller, J. (2006). Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching. *Educational Psychologist, 41*(2), 75-86.

[4] Leinhardt, G., & Ohlsson, S. (1990). Tutorials on the structure of tutoring from teachers. *Artificial Intelligence in Education, 2*(1), 21-46.

[5] Andreasen, R. J., & Wu, C.-H. (1999). Study Abroad Program as an Experiential, Capstone Course: a Proposed Model. 15th Annual Meeting of the Association for International Agricultural and Extension Education, Trinidad & Tobago.

[6] Katz, S., Allbritton, D., & Connelly, J. (2003). Going Beyond the Problem Given: How Human Tutors Use Post-Solution Discussions to Support Transfer. *International Journal of Artificial Intelligence in Education, 13*(1), 79-116.

[7] Katz, S., Connelly, J., & Wilson, C. (2007). Out of the Lab and into the Classroom: An Evaluation of Reflective Dialogue in Andes. In R. Luckin, K. R. Koedinger & J. E. Greer (Eds.), *Artificial Intelligence in Education: Frontiers in Artificial Intelligence and Applications* (Vol. 158, pp. 45-432): IOS Press.

[8] Lee, A. Y., & Hutchison, L. (1998). Improving Learning from Examples through Reflection. *Journal of Experimental Psychology: Applied, 4*(3), 187-210.

[9] Atkins, S., & Murphy, K. (1993). Reflection: A Review of the Literature. *Journal of Advanced Nursing, 18*(8), 1188-1192.

[10] Chi, M. T. H. (2000). Self-Explaining Expository Texts: The Dual Processes of Generating Inferences and Repairing Mental Models. In R. Glasser (Ed.), *Advances in Instructional Psychology: Educational Design and Cognitive Science* (Vol. 5, pp. 161 - 238). Mahwah, NJ: Lawrence Erlbaum Associates.

[11] Kay, J. (1997). *Learner Know Thyself: Student Models to give Learner Control and Responsibility.* Paper presented at the International Conference on Computers in Education, Sarawak, Malaysia.

[12] Mitrovic, A., & Martin, B. (2007). Evaluating the Effect of Open Student Models on Self-Assessment. *Artfiicial Intelligence in Education, 17*(2), 121-144.

[13] Paas, F. G. W. C., & Merrinboer, J. J. G. V. (1993). The Efficiency of Instructional Conditions: An Approach to Combine Mental Effort and Performance Measures. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 35*, 737-743.

[14] Mathews, M., & Mitrovic, A. (2008). *How does students' help-seeking behaviour affect learning?* Paper presented at the 9th International Conference on Intelligent Tutoring Systems, Montreal, Canada.

[15] Mathews, M., Mitrovic, A. & Thomson, D. (2008). *How does students' help-seeking behaviour affect learning?* Analyzing high-level help seeking behaviour in ITSs. W. Nejdl et al. (Eds.): AH 2008, LNCS 5149, pp. 312-315, 2008.